UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/762,828	01/21/2004	Lars Bak	188224/US/3	6090

20686 7590 04/06/2007

DORSEY & WHITNEY, LLP
INTELLECTUAL PROPERTY DEPARTMENT
370 SEVENTEENTH STREET
SUITE 4700
DENVER, CO 80202-5647

EXAMINER	
NGUYEN, PHILLIP H	
ART UNIT	**PAPER NUMBER**
2191	

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	04/06/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

Office Action Summary	Application No.	Applicant(s)	
	10/762,828	BAK ET AL.	
	Examiner	Art Unit	
	Phillip H. Nguyen	2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *21 January 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-23* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-23* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *21 January 2004* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *20040601*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to the original filing date of January 21, 2004, claiming priority date of March 24, 1998, of provisional U.S. Patent Application No. 60/079,765.

2. Claims 1-23 are pending and have been considered below.

Double Patenting

3. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent either is shown to be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

4. Claims 1-23 are rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claims 1-9, 11-23, and 25 of U.S. Patent No. US 6,704,927 B1. Although the conflicting claims are not identical, they are not patentably distinct from each other because the limitations in one claim can obviously be applicable in the corresponding claim.

The following tables show few claims to demonstrate the reason for rejection.

Instant Application No. 10/762,828	US Patent No. 6,704,927 B1
1. A computer-implemented method for increasing the execution performance of a function at run-time, the computer-implemented method comprising: - compiling the function; - identifying a call to a process, the call to the process being included in the function; and - adding dependency information to the function, wherein the dependency information is arranged to indicate a status of the function, the dependency information including class information, name information, and signature information associated with the process, wherein the status of the function is arranged to indicate a validity of the function and a compilation status of the function.	1. A run-time based computer-implemented method for enhancing the execution performance of a function at run-time, the computer-implemented method comprising: - compiling the function at run-time; - identifying a call to a process at run-time, the call to the process being included in the function; and - adding dependency information to the function at run-time, wherein the dependency information is arranged to indicate a status of the function, the dependency information including class information, name information, and signature information associated with the process, wherein the status of the function is arranged to indicate a validity of the function and a

	compilation status of the function.
2. A computer-implemented method as recited in claim 1 wherein the process is a virtual process, the computer-implemented method further including: - analyzing a class structure associated with the function, wherein analyzing the class structure includes determining when the virtual process is a substantially unique target of the call.	2. A computer-implemented method as recited in claim 1 wherein the process is a virtual process, the computer-implemented method further including: - analyzing a class structure associated with the function, wherein analyzing the class structure includes determining when the virtual process is a substantially unique target of the call.
3. A computer-implemented method as recited in claim 2, the computer-implemented method further including: - inlining the virtual process into the function when it is determined that the virtual process is the substantially unique target of the call.	3. A computer-implemented method as recited in claim 2, the computer-implemented method further including: - inlining the virtual process into the function when it is determined that the virtual process is the substantially unique target of the call.
4. A computer-implemented method as	4. A computer-implemented method as

recited in claim 2, the computer-implemented method further including: - placing a direct call to the virtual process in the function.	recited in claim 2, the computer-implemented method further including: - placing a direct call to the virtual process in the function.
5. A computer-implemented method as recited in claim 1 further including: - determining when the function is suitable for compilation.	5. A computer-implemented method as recited in claim 1 further including: - determining when the function is suitable for compilation.
6. A computer-implemented method as recited in claim 1 further including: - loading a class, the class being associated with the function; and - determining when the dependency information indicates that the function is valid, that the function is suitable for deoptimization, and that the function is suitable for reoptimization.	8. A computer-implemented method as recited in claim 1 further including: - loading a class, the class being associated with the function; and 1. A run-time based computer-implemented method for enhancing the execution performance of a function at run-time, the computer-implemented method comprising: - determining whether the compiled function is valid based on the dependency information;

	- determining at least one of whether the function is suitable for deoptimization and whether the function is suitable for reoptimization.
7. A computer-implemented method as recited in claim 6 wherein loading the class includes: - determining when the function is not a substantially unique caller to the process; and - de-compiling the function when is determined that the function is not a substantially unique caller to the process.	6. A computer-implemented method as recited in claim 1 further comprising: - loading a class that is associated with the function; - determining when the function is not a substantially unique caller to the process; and - de-compiling the function when is determined that the function is not a substantially unique caller to the process.
8. A computer-implemented method as recited in claim 6 wherein loading the class includes: - determining when the function is not a substantially unique caller to the process; and	7. A computer-implemented method as recited in claim 1 further comprising: - loading a class that is associated with the function; - determining when the function is not a substantially unique caller to the

- re-compiling the function when is determined that the function is not a substantially unique caller to the process.	process; and - re-compiling the function when is determined that the function is not a substantially unique caller to the process.

Claim Rejections - 35 USC § 101

5. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

6. Claims 15-22 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Regarding claim 15, recites a computer-readable medium that stores the computer codes, which is disclosed as data signal. The specification provides intrinsic evidence that computer-readable medium is intended to cover data signal embodied in a carrier wave (paragraph 0035), such are currently not believed to enable the computer-readable medium to act as a computer hardware component and realizes its functionality absent being claimed in combination with the necessary hardware to receive and convert the data signals to computer codes.

Claims 16-21 directly or indirectly depend on claim 15, and therefore, have been addressed in connection with the same rejection set forth to claim 15.

Regarding claim 22, recites a compiled method structure, which is directed to software, per se, lacking of storage on a medium, which enables any underlying functionality to occur.

Claim Rejections - 35 USC § 112

7. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

8. Claims 2, 3, 7-9, 13, 14, 16, and 17 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Theses claims recite the phrase "substantially" in the body of the claim is unclear to Examiner. The phrase "substantially" is too broad, Examiner does not know how unique is considered as a "substantially" unique.

Claim Rejections - 35 USC § 103

9. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

10. Claims 1-23 are rejected under 35 U.S.C. 103(a) as being unpatentable over Dean et al. (Optimization of Object-Oriented Programs Using Static Hierarchy Analysis,

August 1995, ECOOP 95-Object-Oriented Programming, 9th European Conference), in view of Cabrero et al. (United States Patent No.: 6,260,075).

As per claim 1:

Dean discloses a computer-implemented method for increasing the execution performance of a function at run-time, the computer-implemented method comprising:

- compiling the function (**see at least Introduction, 3rd paragraph**); and
- identifying a call to a process, the call to the process being included in the function (**see at least Introduction, 3rd paragraph**).

Dean does not explicitly discloses:

- adding dependency information to the function, wherein the dependency information is arranged to indicate a status of the function, the dependency information including class information, name information, and signature information associated with the process, wherein the status of the function is arranged to indicate a validity of the function and a compilation status of the function.

However, Cabrero discloses an abstraction layer that allows tasks to share common libraries at runtime. Cabrero's abstraction mechanism proves, among other things, description of service to wrapper functions to perform a requested function (**9:39-11:57**). Therefore, it would have been obvious to one having an ordinary skill in the art at the time the invention was made to combine Cabrero with Dean by using Cabrero teaching of providing the dependency information (i.e., description of services to wrapper

functions) at run-time in Dean because this would ensure proper coordination and state consistency of objects in a multi-user environment.

As per claim 2:

Dean and Cabrero disclose the method in claim 1 above; and Dean further discloses:

- wherein the process is a virtual process, the computer-implemented method further including:
    - analyzing a class structure associated with the function, wherein analyzing the class structure includes determining when the virtual process is a substantially unique target of the call (**see at least section 2.1, 1st paragraph, lines 3-5; page 81, 3rd paragraph, i.e., "A function**...").

As per claim 3:

Dean and Cabrero disclose the method as in claim 2 above; and Dean further discloses:

- inlining the virtual process into the function when it is determined that the virtual process is the substantially unique target of the call (**see at least page 78, 2nd paragraph**).

As per claim 4:

Dean and Cabrero disclose the method as in claim 2 above; and Dean further discloses:

- placing a direct call to the virtual process in the function (**see at least section 2.1, 1st paragraph, line 3-5**).

As per claim 5:

Dean and Cabrero disclose the method as in claim 1 above; and Dean further discloses:

- determining when the function is suitable for compilation (**see at least page 90, 2nd paragraph from bottom, using filtering nodes**).

As per claim 6:

Dean and Cabrero disclose the method as in claim 1 above; and Dean further discloses:

- loading a class, the class being associated with the function (**It is inherent in Dean's approach in order to class hierarchy analysis to be performed**); and
- determining when the dependency information indicates that the function is valid, that the function is suitable for deoptimization, and that the function is suitable for reoptimization (**see at least page 89, 2nd and 3rd paragraphs**).

As per claim 7:

Dean and Cabrero disclose the method as in claim 6 above; and Dean further discloses:

- determining when the function is not a substantially unique caller to the process; and de-compiling the function when is determined that the function is not a substantially unique caller to the process (**see at least page 89, 2nd paragraph**).

As per claim 8:

Dean and Cabrero disclose the method as in claim 6 above; and Dean further discloses:

- determining when the function is not a substantially unique caller to the process (**see at least page 89, 2nd paragraph**); and
- re-compiling the function when is determined that the function is not a substantially unique caller to the process (**see at least page 89, 2nd paragraph**).

As per claim 9:

Dean discloses a computer-implemented method for analyzing a first class associated with a class hierarchy of a system during run-time, the computer-implemented method comprising:

- marking the first class (**see at least Figure on page 84, class A::m and related discussion in the document**);

- marking a second class, the second class being included in the class hierarchy, the second class further being associated with the first class, wherein marking the second class substantially identifies the second class as being associated with the first class (**see at least Figure at page 84, C::m related discussion in the document**);
- inspecting a compiled function associated with the system, the compiled function including dependency information, the dependency information being arranged to indicate a validity status of the compiled function and the optimization status of the compiled function, wherein inspecting the compiled function includes determining when at least one of the first class and the second class is identified in the dependency information (**see at least section 2.2.2, Method Applies-To Sets**); and
- determining when the compiled function is invalid when it is determined that at least one of the first class and the second class is identified in the dependency information (**see at least section 2.2.2 Method Applies-To Sets**).

Dean does not explicitly disclose that the above steps are being performed during run-time. However, Cabrero discloses an abstraction layer that allows tasks to share common libraries at runtime. Cabrero's abstraction mechanism provides, among other things, description of services to wrapper functions to perform a requested function (9:39-11:57). Therefore, it would have been obvious to one having an ordinary skill in the art at the time the invention was made to combine Cabrero with Dean by using Cabrero teaching of providing the dependency information (i.e., description of services

to wrapper functions) at run-time in Dean because this would ensure proper coordination and state consistency of objects in a multi-user environment.

As per claim 10:

Dean and Cabrero disclose the computer-implemented method as in claim 9 above; and further discloses:

- de-compiling the compiled function when it is determined that the compiled function is invalid, wherein de-compiling the compiled function effectively places the compiled function in an interpreted form (**see at least page 89, 2nd and 3rd paragraphs**).

As per claim 11:

Dean and Cabrero disclose the computer-implemented method as in claim 9 above; and further discloses:

- re-compiling the compiled function when it is determined that the compiled function is invalid, wherein re-compiling the compiled function allows the compiled function to account for the first class (**see at least page 89, 2nd and 3rd paragraphs**).

As per claims 12-14:

- system claims, recite the same limitations as recited in claims 1-3 respectively, and therefore, have been addressed in connection with the rejection set forth to claims 1-3 respectively.

As per claims 15-20:

- computer program product claims, recite the same limitations as recited in claims 1-6 respectively, and therefore, have been addressed in connection with the same rejection set forth to claim 1-6 respectively.

As per claim 21:

Dean and Cabrero disclose the computer program product as in claim 15; and Cabrero further discloses:

- wherein the computer-readable medium is one selected from the group consisting of a floppy disk, a hard disk, a tape, a data signal embodied in a carrier wave, a CD-ROM, a system memory, and a flash memory (**4:24-42**).

As per claims 22 and 23:

- recites the same limitations as recited in claim 1 above, and therefore, have been addressed in connection with the rejection set forth to claim 1.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Phillip H. Nguyen whose telephone number is (571) 270-1070. The examiner can normally be reached on Monday - Thursday 10:00 AM - 3:00 PM EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Y. Zhen can be reached on (571) 272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

PN
03/15/2007

WEI ZHEN
SUPERVISORY PATENT EXAMINER